

Claims

We claim:

1. A computer-implemented method for creating a graphical program, the
5 method comprising:

creating a first graphical program and storing the first graphical program in a
memory;

associating a debugging graphical program at a debugging location in the first
graphical program, wherein said associating does not modify the first graphical program;

10 wherein the debugging graphical program is executable during execution of the
first graphical program to aid in debugging at least a portion of the first graphical
program.

2. The method of claim 1, wherein said associating does not require a re-
15 compilation of the first graphical program.

3. The method of claim 1, further comprising:
executing the first graphical program up to the debugging location;
executing the debugging graphical program after executing the first graphical
20 program up to the debugging location; and

the debugging graphical program generating debugging results, wherein the
debugging results are useful in analyzing at least a portion of the first graphical program.

4. The method of claim 3, further comprising:
25 completing execution of the first graphical program based on the debugging
results of said executing the debugging graphical program.

5. The method of claim 3,

wherein said executing the debugging graphical program includes displaying the debugging results of the debugging graphical program.

6. The method of claim 3, wherein said executing the debugging graphical program comprises:

receiving data from the first graphical program; and
performing one or more of:

displaying the data from the first graphical program; and/or
logging the data from the first graphical program to a file.

7. The method of claim 3, wherein said executing the debugging graphical program comprises:

receiving data from the first graphical program;
generating statistics based on the received data; and
displaying the statistics.

8. The method of claim 7, wherein said statistics comprise one or more of:
data generated by the debugging graphical program;
data generated by a plurality of executions of the debugging graphical program during a corresponding plurality of executions of the first graphical program, wherein said data generated by the plurality of executions of the debugging graphical program includes differences in execution times between the plurality of executions of the debugging graphical program, wherein said differences in execution times are useable in optimizing performance the main graphical program.

9. The method of claim 3,
wherein said completing execution of the first graphical program is performed in single stepping mode based on the debugging results of said executing the debugging graphical program.

10. The method of claim 1, further comprising:
executing the first graphical program up to the debugging location, wherein the
first graphical program generates data at the debugging location;
5 providing the data to the debugging graphical program;
executing the debugging graphical program, wherein the debugging graphical
program uses the data;
the debugging graphical program generating debugging results;
based on the debugging results, performing one or more of: halting execution of
10 the first graphical program; entering single stepping mode in the first graphical program;
or completing execution of the first graphical program.

11. The method of claim 10, wherein the first graphical program executes up
to the debugging location where the debugging graphical program is associated, and waits
15 for user input.

12. The method of claim 1,
wherein the first graphical program comprises a plurality of interconnected nodes,
which visually indicate functionality of the first graphical program.
20

13. The method of claim 12,
wherein the first graphical program comprises a plurality of data flow paths;
wherein said associating a debugging graphical program at a location in the first
graphical program comprises associating the debugging graphical program at a first data
25 flow path in the first graphical program.

14. The method of claim 13, wherein said associating comprises:
storing information in at least one data structure, wherein the information
comprises information regarding the first graphical program, the debugging graphical

program, and the location where the debugging graphical program is attached along the first data flow path of the first graphical program.

15. The method of claim 13, wherein said associating comprises:
5 receiving user input from a pointing device selecting a data flow path in the first graphical program;
display a plurality of debugging graphical programs; and
receiving user input selecting one of the debugging graphical programs.

10 16. The method of claim 13, wherein said associating comprises:
receiving user input selecting a data flow path in the first graphical program,
wherein the data flow path is configured to carry data of a first data type;
determining the first data type of the data flow path
display a plurality of debugging graphical programs appropriate for the first data
15 type of the data flow path.

17. The method of claim 13, wherein said associating the debugging graphical program at the debugging location in the first graphical program comprises associating the debugging graphical program at a node in the first graphical program.

20 18. The method of claim 1, further comprising:
disassociating the debugging graphical program from the first graphical program,
wherein said disassociating does not modify the first graphical program and/or does not require a re-compilation of the first graphical program.

25 19. The method of claim 1,
wherein the first graphical program is located on a first computer system;

wherein the debugging graphical program is located on a second computer system, wherein the second computer system is coupled to the first computer system over a network.

5 20. The method of claim 19, the method further comprising:
 executing the first graphical program on the first computer system up to the debugging location;

 executing the debugging graphical program on the second computer system, wherein the debugging graphical program is executed after executing the first graphical
10 program on the first computer system up to the debugging location;

 the debugging graphical program generating debugging results on the second computer system; and

 providing the debugging results from the second computer system to the first computer system.

15 21. The method of claim 1,
 wherein the first graphical program is located on a first computer system, wherein the first computer system is a target computer system coupled to or comprised in a second computer system;

20 wherein the debugging graphical program is located on and executed on the first computer system.

 22. The method of claim 1,
 wherein the first graphical program is located on a first computer system, wherein
25 the first computer system is a target computer system coupled to or comprised in a second computer system;

 wherein the debugging graphical program is located on and executed on the second computer system.

23. A computer-implemented method for executing a first graphical program, the method comprising:

executing the first graphical program up to a debugging location, wherein the first graphical program generates data at the debugging location;

providing the data to a debugging graphical program;

executing the debugging graphical program, wherein the debugging graphical program uses the data;

the debugging graphical program generating debugging results;

wherein use of the debugging graphical program does not require modification or recompilation of the first graphical program.

24. The method of claim 23,

after the debugging graphical program generates debugging results, performing one or more of: halting execution of the first graphical program; entering single stepping mode in the first graphical program; or completing execution of the first graphical program.

25. The method of claim 23, further comprising:

associating the debugging graphical program at the debugging location in the first graphical program;

wherein said associating does not require modification or recompilation of the first graphical program.

26. The method of claim 23,

wherein the first graphical program is located on a first computer system;

wherein the debugging graphical program is located on a second computer system, wherein the second computer system is coupled to the first computer system over a network;

wherein the first graphical program executes on the first computer system up to the debugging location;

wherein the debugging graphical program executes on the second computer system, wherein the debugging graphical program is executed after executing the first graphical program on the first computer system up to the debugging location; and

wherein the debugging graphical program generates debugging results on the second computer system.

27. A computer-implemented method for analyzing a first graphical program, the method comprising:

storing the first graphical program in a memory of the computer;

associating a second graphical program at a location in the first graphical program, wherein said associating does not modify the first graphical program;

wherein the second graphical program is executable during execution of the first graphical program to aid in analyzing at least a portion of the first graphical program.

28. The method of claim 27, wherein said associating does not require a re-compilation of the first graphical program.

29. The method of claim 27, further comprising:

executing the first graphical program up to the location;

executing the second graphical program after executing the first graphical program up to the location; and

the second graphical program generating results, wherein the results are useful in analyzing at least a portion of the first graphical program.

30. The method of claim 27, further comprising:

executing the first graphical program up to the location, wherein the first graphical program generates data at the location;

providing the data to the second graphical program;

executing the second graphical program, wherein the second graphical program
5 uses the data;

the second graphical program generating results;

based on the results, performing one or more of: halting execution of the first graphical program; entering single stepping mode in the first graphical program; or completing execution of the first graphical program.

10

31. A memory medium comprising program instructions for analyzing a first graphical program, wherein the program instructions are executable to implement:

storing the first graphical program in a memory of the computer;

15 associating a second graphical program at a location in the first graphical program, wherein said associating does not modify the first graphical program;

wherein the second graphical program is executable during execution of the first graphical program to aid in analyzing at least a portion of the first graphical program.

20 32. The memory medium of claim 31, wherein said associating does not require a re-compilation of the first graphical program.

33. The memory medium of claim 31, wherein the program instructions are further executable to implement:

25 executing the first graphical program up to the location;

executing the second graphical program after executing the first graphical program up to the location; and

the second graphical program generating results, wherein the results are useful in analyzing at least a portion of the first graphical program.

34. The memory medium of claim 31, wherein the program instructions are further executable to implement:

executing the first graphical program up to the location, wherein the first graphical program generates data at the location;

providing the data to the second graphical program;

executing the second graphical program, wherein the second graphical program uses the data;

the second graphical program generating results;

based on the results, performing one or more of: halting execution of the first graphical program; entering single stepping mode in the first graphical program; or completing execution of the first graphical program.

35. A memory medium which stores:

a first graphical program, wherein the first graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the first graphical program;

a second graphical program;

a data structure which is operable to store information associating the second graphical program with a location in the first graphical program, wherein the first graphical program is not modified by the second graphical program;

wherein the second graphical program is executable during execution of the first graphical program to aid in analyzing at least a portion of the first graphical program.

36. The memory medium of claim 35,

where the first graphical program executes up to the location, wherein the first graphical program generates data at the location;

providing the data to the second graphical program;

executing the second graphical program, wherein the second graphical program uses the data;

the second graphical program generating results;

based on the results, performing one or more of: halting execution of the first graphical program; entering single stepping mode in the first graphical program; or
5 completing execution of the first graphical program.

FOR 2020-09-04 10:54:05